Derandomizing Random Walks in Undirected Graphs Using Locally Fair Exploration Strategies

Colin Cooper, David Ilcinkas, Ralf Klasing, Adrian Kosowski

Derandomizing Random Walks in Undirected Graphs Using Locally Fair Exploration Strategies *

Colin Cooper[1], David Ilcinkas[2], Ralf Klasing[2], and Adrian Kosowski[2,3]

[1] Dept of Computer Science, King's College London
colin.cooper@kcl.ac.uk
[2] LaBRI, CNRS and Université de Bordeaux
{ilcinkas, klasing}@labri.fr
[3] Dept of Algorithms and System Modeling, Gdańsk University of Technology
adrian@kaims.pl

Abstract. We consider the problem of exploring an anonymous undirected graph using an oblivious robot. The studied exploration strategies are designed so that the next edge in the robot's walk is chosen using only local information, and so that some local equity (fairness) criterion is satisfied for the adjacent undirected edges. Such strategies can be seen as an attempt to derandomize random walks, and are natural undirected counterparts of the rotor-router model for symmetric directed graphs.
The first of the studied strategies, known as Oldest-First (OF), always chooses the neighboring edge for which the most time has elapsed since its last traversal. Unlike in the case of symmetric directed graphs, we show that such a strategy in some cases leads to exponential cover time. We then consider another strategy called Least-Used-First (LUF) which always uses adjacent edges which have been traversed the smallest number of times. We show that any Least-Used-First exploration covers a graph $G = (V, E)$ of diameter D within time $O(D\,|E|)$, and in the long run traverses all edges of G with the same frequency.

1 Introduction

A widely studied problem concerns the exploration of an anonymous graph $G = (V, E)$, with the goal of visiting all its vertices and regularly traversing its edges. At each discrete moment of time, the robot is located at a node of the graph, and is provided with only a local view of the adjacent edges of the graph. The exploration strategies studied in this paper fall into the line of research devoted to derandomizing random walks in graphs [4, 7, 19, 20, 22].

The *random walk* is an oblivious exploration strategy in which the edge used by the robot to exit its current location is chosen with equal probability from

* The research was partially funded by the KBN (Poland) Grant 4 T11C 047 25, by the ANR projects "ALADDIN" and "ALPAGE", by the INRIA project "CEPAGE", and by European COST Action Action 295 "Dynamic Communication Networks" (DYNAMO).

among all the edges adjacent to the current node; cf. e.g. [1,16] for an extensive introduction to the topic. Explorations achieved through random walks are on average good, in the sense that the following properties hold *in expectation*:

(1) Within polynomial time, the walk visits all of the vertices of the graph.
(2) Within polynomial time, the walk stabilizes to the steady state, and henceforth all edges are visited with the same frequency.

We focus on the problem of designing local exploration strategies which derandomize a random walk in a graph in an attempt to achieve the above stated properties in the deterministic sense of *worst-case performance*. The next vertex to be visited should depend only on the values of certain parameters associated with the edges adjacent to the current node. Such a problem naturally gives rise to the definition of *locally equitable strategies*, i.e. strategies, in which at each step the robot chooses from among the adjacent edges the edge which is in some sense the "poorest", in an effort to make the traversal fair. In this context, two natural notions of equity may be defined:

– An exploration is said to follow the *Oldest-First* (OF) strategy if it directs the robot to an unexplored neighboring edge, if one exists, and otherwise to the neighboring edge for which the most time has elapsed since its last traversal, i.e. the edge which has waited the longest.
– An exploration is said to follow the *Least-Used-First* (LUF) strategy if it directs the robot to a neighboring edge which has so far been visited by the robot the smallest number of times.

When the considered graph is *symmetric and directed*, and the above definitions are applied to directed edges, then the Oldest-First notion of equity is known to be strictly stronger than Least-Used-First, i.e. any exploration which follows the OF strategy also follows the LUF strategy [22]. Moreover, the Oldest-First strategy is in this context equivalent to a well-established efficient exploration model based on the rotor-router model (a.k.a. the "Propp machine", cf. e.g. [5] for an introduction of the model). In the directed case, both of the described locally fair exploration stratagies are known to preserve properties (1) and (2) of the random walk. More precisely, for a symmetric directed graph of diameter D, any exploration which follows such a strategy achieves a *cover time* of $O(D\,|E|)$ and stabilizes to a globally fair traversal of all the edges. Herein we look at the Oldest-First and Least-Used-First strategies when applied to the *undirected* edges of a graph. For this case, the results, and the used techniques, turn out to be surprisingly different.

Basic parameters. Two parameters of interest when discussing exploration strategies are the *cover time* of a graph and the *traversal frequency* of its edges. We introduce them first in the context of random walks.

Let $\mathbf{C}_s$ be the random variable describing the number of steps required for a random walk starting at vertex s, to visit every vertex of the graph. Then the *cover time* of the graph is the maximum, taken over all starting vertices s, of the expected values of variables $\mathbf{C}_s$, $\mathcal{C}(G) = \max_{s\in V} \mathbf{E}\ \mathbf{C}_s$. Let $\mathbf{c}_{s,e}(t)$ be the random variable describing the number of visits to edge e within time t, for a random walk starting at vertex s. We can define random variables describing the distribution of visits to edges for sufficiently large time, $\mathbf{f}_{s,e} = \liminf_{t\to\infty} \mathbf{c}_{s,e}(t)/t$ (where lim inf is used instead of lim to guarantee correctness of the definition). The *traversal frequency* $f_e(G)$ of an edge e is defined as the minimum, taken over all starting vertices s, of the expected values of variables $\mathbf{f}_{s,e}$, $f_e(G) = \min_{s\in V} \mathbf{E}\ \mathbf{f}_{s,e}$.

Given any exploration algorithm $\mathcal{E}$ which is fully deterministic (or in other words, a specific exploration), the notions of *cover time for* $\mathcal{E}$ and *traversal frequency for* $\mathcal{E}$ can be defined analogously. The only difference is that then the variables $\mathbf{C}_s$ and $\mathbf{f}_{s,e}$ are deterministically defined, hence we need not speak of their expected values.

Related work. We confine ourselves to a short survey of works on random walks, and the rotor-router model and its variants. Many other approaches to the derandomization of random walks have been studied, most notably, through universal traversal sequences [2] (UTS) and universal exploration sequences [15] (UXS). UTS-s can be constructed in polylogarithimic space using pseudorandom generators, cf. e.g. [18], whereas UXS-s have been proved to be constructible in log-space [17].

Exploration with random walks. In expectation, random walks quickly "hit" all vertices, and the cover time $\mathcal{C}(G)$ of a connected graph satisfies the inequalities $\mathcal{C}(G) \geq |V| \log |V|$ and $\mathcal{C}(G) = O(|V|^3)$ [2]. With respect to the diameter, the cover time is upper bounded by $O(D\,|E| \log |V|)$. In fact, for many special graph classes, such as complete graphs, expanders, trees, or grids, tighter bounds on cover time can be obtained [1].

Random walks directly capture the property of equity in the sense that, for a random walk in the steady state, the expected frequency of visits to each edge is the same. More precisely, for a random walk on a connected undirected non-bipartite graph G, the stationary distribution of visits to edges is the uniform distribution with parameter $1/|E|$, thus for any e, $f_e(G) = 1/|E|$. Similarly, if we replace each edge $\{u, v\}$ with two symmetric directed edges (u, v), (v, u) then the stationary distribution of visits is again uniform with parameter $1/(2|E|)$, and so for any directed edge $\boldsymbol{e}$, $f_{\boldsymbol{e}}(G) = 1/(2|E|)$.

In expectation, the random walk stabilizes to such a fair traversal of the edges very quickly. Several notions have been introduced, informally corresponding to the expected moment at which (for a regular graph) all vertices have been visited a similar number of times, cf. [21]. One of the most studied is that of blanket time, which has been shown to be within a factor of $O(\log \log |V|)$ of the cover time, for all graphs [12].

Equitable exploration of directed graphs. For symmetric directed graphs, the Oldest-First exploration strategy corresponds to exploration in the rotor-router model, i.e. a set-up in which edges exiting each node have successive labels, and the next edge to be traversed is selected by a pointer. After this edge is traversed, the pointer moves on to the edge with the next label, in a cyclic way. This approach was first studied in [4, 19, 20], and the cover time of Oldest-First for directed graphs was shown to be $O(|V||E|)$. Slightly later [22] obtained an improved bound on cover time of $O(D\,|E|)$, and also showed that after time at most $O(D\,|E|)$ the exploration stabilizes to a periodic traversal of some directed Eulerian cycle of the graph (containing each directed edge exactly once, i.e. of length $2|E|$). Consequently, Oldest-First explorations on symmetric directed graphs are fair, in the sense that all edges are visited with the same frequency $f_{\boldsymbol{e}}(G) = 1/(2|E|)$.

When considering symmetric directed graphs, an exploration achieved in accordance with the Oldest-First rule also satisfies the conditions of a Least-Used-First exploration. Whereas a Least-Used-First exploration need not in general stabilize to a traversal of a directed Eulerian cycle, it also retains the property that for any time moment, the number of visits to any two edges outgoing from the same vertex can differ by at most 1 [13, 14]. This property immediately implies that for symmetric directed graphs, any execution of Least-Used-First has a cover time of $O(D\,|E|)$, and also visits all directed edges with the same frequency.

In a slightly wider context, local exploration strategies have been considered for robots with bounded memory, cf. e.g. [8, 9, 17]. In some settings, the robot is additionally assisted by identifiers or markers placed on the nodes and/or edges of the explored graph, cf. e.g. [3, 6, 10].

Our results. Herein we establish certain properties of explorations which follow the Oldest-First or Least-Used-First strategies in undirected graphs.

*The Oldest-First (*OF*) strategy* in undirected graphs can be regarded as a natural analogue of the Oldest-First strategy (rotor-router model) for symmetric directed graphs. However, whereas the rotor-router model leads to explorations which traverse directed edges with equal frequency, and have a cover time bounded by $O(D\,|E|)$, this is not the case for Oldest-First explorations in undirected graphs. Indeed, in Section 2 we show the following theorems.

– In some classes of undirected graphs, any exploration which follows the Oldest-First strategy is unfair, with an exponentially large ratio of visits between the most often and least often visited edges (Theorem 1).
– There exist explorations following the Oldest-First strategy which have exponential cover time of $2^{\Omega(|V|)}$ in some graph classes (Theorem 2).

*The Least-Used-First (*LUF*) strategy* in undirected graphs is fundamentally better than the Oldest-First strategy, which is contrary to the situation in symmetric directed graphs. In fact, in Section 3 we show that, in undirected graphs, explorations which follow the LUF strategy are fair, efficient, and tolerant to perturbations of initial conditions, as expressed by the following theorems.

– Any exploration of an undirected graph which follows the Least-Used-First strategy is fair, achieving uniform distribution of visits to all edges (Theorem 5).
– Any exploration of an undirected graph which follows the Least-Used-First strategy achieves a cover time of $O(D\,|E|)$, where D denotes the diameter (Theorem 4). This bound is tight (Theorem 3). When the exploration starts from a state with non-zero (corrupted) initial values of traversal counts on edges, the cover time is bounded by $O((|V|+p)|E|)$, where p is the maximal value of a counter in the initial state (Theorem 6).

Notation. Unless otherwise stated, all considered graphs are assumed to be simple, undirected, and connected. The explored graph is denoted by $G = (V, E)$, with $|V| = n$ and $|E| = m$. The diameter of the graph is denoted by D and its maximum vertex degree by Δ. The set of neighbors of a vertex $v \in V$ is denoted by N_v. The set of non-negative integers is denoted by $\mathbb{N}$. A discrete interval $[a, b]$ is defined as the set of all integers k such that $a \le k \le b$ ($[a, b] = \emptyset$ when $a > b$).

2 The Oldest-First (OF) Strategy

In this section we show that any OF exploration is unfair (Theorem 1), and moreover that OF explorations may sometimes take exponential time to cover the whole graph (Theorem 2).

Theorem 1. *There exists a family of graphs $(G_n)_{n\ge1}$ of order $\Theta(n)$, such that for each graph G_n in this family, some two of its edges e and e' satisfy $\frac{f_e(G_n)}{f_{e'}(G_n)} = (\frac{3}{2})^n$ with $f_{e'}(G_n) \neq 0$, for any exploration following the OF strategy.*

Proof. Fix an arbitrary positive integer n. Let G_n be the graph defined as follows. The nodes are denoted $v_j^{(k)}$, for any $j \in [1, 7]$ and any $k \in [1, n]$. Moreover, we have that $v_7^{(k)} = v_1^{(k+1)}$ for any $k \in [1, n-1]$. This means that G_n has $6n+1$ nodes. The $8n$ edges are the following: $e_1^{(k)} = \{v_1^{(k)}, v_2^{(k)}\}$, $e_2^{(k)} = \{v_2^{(k)}, v_3^{(k)}\}$, $e_3^{(k)} = \{v_2^{(k)}, v_4^{(k)}\}$, $e_4^{(k)} = \{v_3^{(k)}, v_5^{(k)}\}$, $e_5^{(k)} = \{v_4^{(k)}, v_5^{(k)}\}$, $e_6^{(k)} = \{v_2^{(k)}, v_6^{(k)}\}$, $e_7^{(k)} = \{v_5^{(k)}, v_6^{(k)}\}$, and $e_8^{(k)} = \{v_6^{(k)}, v_7^{(k)}\}$, for any $k \in [1, n]$. The graph G_n is depicted in Figure 1.

We assume that the exploration is starting from $v_1^{(1)}$. We will now focus on a block B of G_n, that is on the subgraph of G_n induced by the 7 nodes

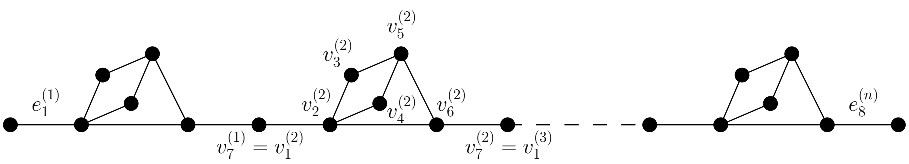

Fig. 1. The graph G_n.

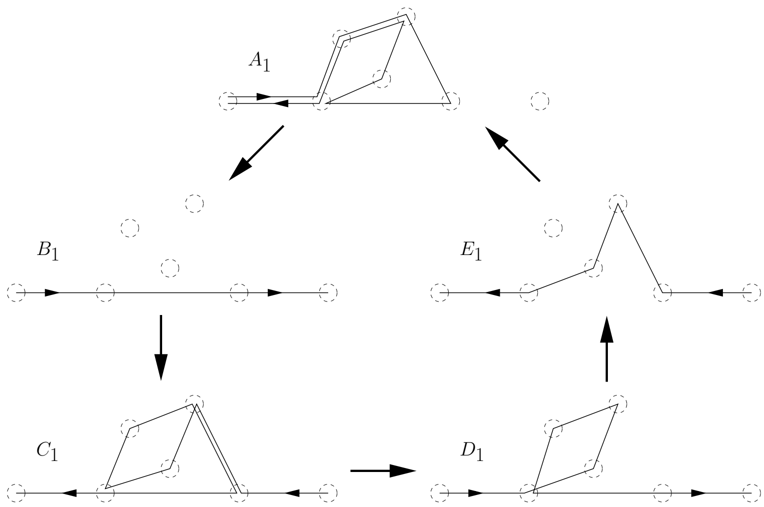

Fig. 2. The two possible cycles of traversals of a block B. Cycle $(A_1, B_1, C_1, D_1, E_1)$ is presented in the figure. Cycle $(A_2, B_2, C_2, D_2, E_2)$ is obtained as follows: $A_2 = \bar{A}_1$, $B_2 = \bar{E}_1$, $C_2 = \bar{D}_1$, $D_2 = \bar{C}_1$, $E_2 = \bar{B}_1$, where $\bar{X}$ denotes the reversal of the direction of the exploration route in X.

$\{v_1^{(k)}, \cdots, v_7^{(k)}\}$, for an arbitrary and fixed $k \in [1, n]$. To simplify the notation, we will remove the superscript $^{(k)}$ in the following, when there are no ambiguities.

There may be several different explorations following the OF strategy from $v_1^{(1)}$. Indeed, when the exploration reaches a node with at least two edges that are not yet explored, the exploration may proceed along any of these unexplored edges.

By a tedious case-by-case analysis, we show that the behavior of the robot in successive traversals of a given block follows a cyclic pattern, as shown in Figure 2. In all the cases, in the time period during which the edge e_8 is traversed 4 times, the edge e_1 is traversed 6 times. We now notice that the exploration becomes eventually periodic. Indeed, only the local ordering of the last traversal times of the incident edges at each node influences the exploration. Therefore the number of different possible configurations of the graph and its ongoing exploration is bounded by some (large) function of n. Therefore, the exploration is eventually periodic and, for any edge e of the graph, the sequence $c_e(t)/t$ converges to the actual frequency of traversals $f_e(G_n)$ of the edge e. In particular, we have $\sum_{e \in E(G_n)} f_e(G_n) = 1$. Since we just proved that for any $k \in [1, n]$ we have $f_{e_1^{(k)}}(G_n) = \frac{3}{2} f_{e_8^{(k)}}(G_n)$, we have $f_{e_1^{(1)}}(G_n) = (\frac{3}{2})^n f_{e_8^{(n)}}(G_n)$, with $f_{e_8^{(n)}}(G_n) \neq 0$. This concludes the proof of the theorem. □

Theorem 2. *There exists a family of graphs $(G_n)_{n \geq 1}$ of order $\Theta(n)$, such that for each graph G_n in this family, some exploration following the* OF *strategy has a cover time of $2^{\Omega(n)}$.*

Proof. We consider the family of graphs described in Theorem 1. Given an arbitrary execution $\mathcal{E}$ of the OF strategy, there exist two edges e and e' satisfying $\frac{f_e(G)}{f_{e'}(G)} = (\frac{3}{2})^n$ (with $f_{e'}(G) \neq 0$). Therefore, there exist two times t_1 and t_2, with $t_2 - t_1 \geq (\frac{3}{2})^n - 1$, such that the edge e' is not traversed between time t_1 and t_2. Let v be the current position of the traversal $\mathcal{E}$ at time t_1. Then, consider the exploration $\mathcal{E}'$ which starts at v and has the same execution from the beginning, as $\mathcal{E}$ from time t_1. It is clear that $\mathcal{E}'$ follows the OF strategy, and moreover it will not traverse e' before time $t_2 - t_1$. Thus, $\mathcal{E}'$ has a cover time of at least $(\frac{3}{2})^n - 1$. $\square$

3 The Least-Used-First (LUF) Strategy

In Subsection 3.2 we will show that LUF strategies are fair and cover any graph in $O(m\,D)$ time. Before doing this, in Subsection 3.1 we construct a family of examples showing that such a bound on cover time is essentially tight.

3.1 A Worst Case Lower Bound on Cover Time

Theorem 3. *For sufficiently large n, $m \in [n-1, n(n-1)/2]$ and $D \leq n$, the worst-case cover time of the LUF strategy in the family of graphs of at most n nodes, at most m edges, and diameter at most D, is $\Omega(m\,D)$.*

Proof. Fix $n \geq 16$, $m \in [n-1, n(n-1)/2]$ and $D \in [8, n]$. Let G be the graph defined as follows, see Figure 3. Let $n_C = \lfloor D/8 \rfloor$. The graph G first consists of $3n_C + 1$ nodes organized in a chain of 4-node cycles. Let n_K be the largest even integer smaller than $n/2$ such that $n_K(n_K+1)/2 < m/2$. The graph G also consists of n_K additional nodes forming together with one extreme node of the chain a complete graph on n_K+1 vertices. To summarize, G has $n_K+3n_C+1 \leq n$ nodes, $4n_C + n_K(n_K+1)/2 \leq m$ edges and diameter $2n_C + 1 \leq D$.

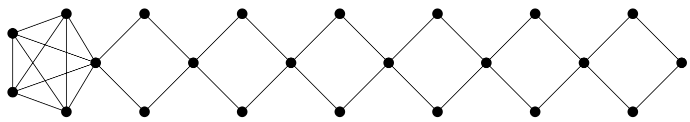

Fig. 3. The graph G with $n_C = 6$ and $n_K = 4$.

It is easily shown that the worst-case cover time of G is at least $n_K(n_K + 1)/2 \cdot n_C$; we leave out the details of the analysis. Since $n_K(n_K+1)/2 \in \Omega(m)$ and $n_C \in \Omega(D)$, the theorem holds. $\square$

3.2 An Upper Bound on Cover Time

We now proceed to prove the $O(m\,D)$ bound on cover time of any LUF exploration, through a sequence of technical lemmas.

Throughout the proofs we will use the following notation. When describing moments of time, the symbol t' is treated as a more compact notation for $t+1$, likewise t'' means $t+2$. The vertex occupied by the robot at time t is denoted by $r(t)$; the starting vertex of exploration is denoted by s, that is $s = r(0)$. With each edge e we associate a counter c_e called its *traversal count*, whose value at time t is denoted by $c_e(t)$; initially we assume $c_e(0) = 0$ for all $e \in E$. When traversing edge e in the time interval (t, t') we only increment the value of the counter associated with this edge, $c_e(t') = c_e(t) + 1$. For each node u we denote by $C_u(t)$ the set of traversal counts of the adjacent edges at time t: $C_u(t) = \{c_{\{u,v\}}(t) : v \in N_u\}$. The set of traversal counts of all edges of the graph is denoted by $C(t) = \{c_e(t) : e \in E\}$.

At any given time t, let parameter $k \in \mathbb{N} \cup \{-1\}$ be defined in such a way that $\max C(t) \in [2k+1, 2k+2]$, and let parameter $l \in \mathbb{N}$ be such that $\min C_r(t) \in [2l, 2l+1]$. Parameters k, l, and r used without an indication of time are assumed to refer to the moment of time denoted by t, while symbols k', l', r', and r'' should be treated as equivalent to $k(t')$, $l(t')$, $r(t')$, and $r(t'')$, respectively.

We start by making the following claim which is a simple extension of the following observation: for each vertex v different from both r and s, the total number of traversals of edges incident to v, performed when entering v, is the same as the total number of traversals of these edges performed when leaving v.

Lemma 1. *For a node $u \in V$, let $S_u(t) = \sum_{v \in N_u} c_{\{u,v\}}(t)$. If $S_u(t)$ is odd, then $r \neq s$ and either $u = r$, or $u = s$.*

Lemma 2. *If for some time moment t we have $k' = k+1$, then $r = s$ and $C_s(t) = \{2k+2\}$.*

Proof. If $k' = k+1$, then clearly $c_{\{r,r'\}}(t) = 2k+2 = \max C(t)$. This implies that during the time interval (t, t') the robot chooses an edge having the maximal traversal count. Clearly, this means that there is no edge with a smaller traversal count available at r, so $C_r(t) = \{2k+2\}$. Hence, in Lemma 1 the value of $S_r(t)$ is even, and we immediately obtain the claim, $r = s$. □

Lemma 3. *For any time moment t, $\max C_s(t) \geq 2k+1$.*

Proof. We can obviously assume that $k \geq 0$. Let $\tau < t$ be such a time moment that $k(\tau) = k-1$ and $k(\tau') = k$. Then by Lemma 2, $r(\tau) = s$ and $C_s(\tau) = \{2(k-1)+2\} = \{2k\}$. So, after traversing any edge adjacent to s, we obtain $\max C_s(\tau') = 2k+1$. Since $t \geq \tau'$ and $\max C_s(t) \geq \max C_s(\tau')$, the claim follows directly. □

Lemma 4. *If for some time moment t we have $C_r(t) = \{2p+2\}$, where p is some integer, then $r = s$ and $p = k$.*

Proof. When $C_r(t) = \{2p+2\}$, in Lemma 1 the value of $S_r(t)$ is even, and so $r = s$. Moreover, by Lemma 3 we cannot have $p < k$ since then $\max C_s(t) \leq 2k$. Thus $p = k$. □

Lemma 5. *For any time moment t, the following statements hold:*

- *there exists a subset $V_A = \{v_l, ..., v_{k-1}\}$ of vertices indexed by integers $a \in [l, k-1]$, such that $C_{v_a}(t) \subseteq [2a, 2a+3]$.*
- *for any other vertex $v \notin V_A$ we have $C_v(t) \subseteq [2b, 2b+2]$ for some integer value b.*

Proof. Initially, for $t = 0$ we have $C(0) = \{0\}$, $k = -1$, $l = 0$, and so the induction claim holds with $V_A(0) = \emptyset$.

Assuming that the induction assumption holds for time t and a corresponding set V_A is given, we will now prove that it also holds for time t' with an appropriately modified set V'_A. (Sometimes no modification will be necessary; for example, when G is a cycle, we have $V_A(\tau) = \emptyset$ for all $\tau \geq 0$.) We start by showing a small auxiliary claim.

Claim. $l' \in [l-1, l+1]$.

Proof: The traversal count, directly before traversal, of the edge used in time interval (t', t'') can be greater by at most one than that of the edge used in time interval (t, t'), so $c_{\{r', r''\}}(t') \leq c_{\{r,r'\}}(t) + 1 \leq 2l + 2$, and thus $l' \leq l+1$. Suppose that $l' < l$; then we have $\min C_{r'}(t) < 2l$, and by the inductive assumption $r' \notin V_A(t)$. Thus $\max C_{r'}(t) - \min C_{r'}(t) \leq 2$, and we obtain $c_{\{r',r''\}}(t') = \min C_{r'}(t') \geq \min C_{r'}(t) \geq \max C_{r'}(t) - 2 \geq 2l - 2$, which means that always $l' \geq l - 1$, completing the proof of the claim.

Now, consider the following definition of set $V'_A = \{v'_a : a \in [l', k'-1]\}$ for time t': (1) For all $a \in [l+1, k-1]$, put $v'_a := v_a$; (2) If $l' \leq l$ and $l' < k'$, put $v'_l := v_l$; (3) If $l' = l - 1$ and $l' < k'$, put $v'_{l-1} := r'$. The above procedure clearly defines all elements v'_a for $a \in [l', k-1]$. We now observe that it does in fact define all elements v'_a for the whole of the required range, $a \in [l', k'-1]$. Indeed, if $k' = k+1$, by Lemma 2 we have $c_{\{r,r'\}}(t) = 2k+2$, so $l = k+1 = k'$. Consequently, if $l' \geq k+1$ in the proposed construction, then set V'_A is empty as required, and if $l' = l - 1 = k$, then the only element v'_{l-1} of V'_A is well defined.

We now verify the induction claim for the proposed definition of set V'_A by checking the imposed bounds on sets $C_v(t')$, for all vertices $v \in V$. Taking into account that for all vertices v other than r and r' we have $C_v(t) = C_v(t')$, by the construction of elements v'_a based on elements v_a, it is evident that it now suffices to check the bounds on $C_v(t')$ for $v \in \{r, r', v_l\}$; for all other vertices, the bounds follow directly from the induction assumption for time t. We therefore now successively consider vertices r, r', and v_l.

For vertex r we need to consider two possibilities: either $r \in V_A$, or $r \notin V_A$.

1. If $r \in V_A$, then $V_A \neq \emptyset$ and so $l \leq k-1$. Since $\min C_r(t) \in [2l, 2l+1]$ by the definition of l, taking into account the inductive assumption concerning the bounds on $C_r(t)$ we must have $r = v_l$ (note that vertices v_a are only defined for indices $a \geq l$) and $C_r(t) \subseteq [2l, 2l+3]$. After traversing edge $\{r, r'\}$, we have $c_{\{r,r'\}}(t') = c_{\{r,r'\}}(t) + 1 = \min C_r(t) + 1 \in [2l+1, 2l+2]$, so we retain the property $C_r(t') \subseteq [2l, 2l+3]$. If $r = v'_l$, the bounds on set $C_r(t')$ are thus satisfied. We will now show that the other case, $r \neq v'_l$, is impossible. Indeed, when $r \neq v'_l$ we would have $l' = l+1$ (otherwise, $l' \leq l$ would mean

that $l' \leq l < k \leq k'$, so $v'_l = v_l = r$). Therefore, $\min C_{r'}(t') \geq 2l+2$, so $c_{\{r,r'\}}(t') = 2l+2$ and $c_{\{r,r'\}}(t) = 2l+1$. Taking into account that $r' \neq r = v_l$, we have $r' \notin V_A$ (as $\min C_{r'}(t) \leq 2l+1$) and $C_{r'}(t) \subseteq [2l, 2l+2]$. As we have already observed that $\min C_{r'}(t') \geq 2l+2$ and $c_{\{r,r'\}}(t') = 2l+2$, we obtain $C_{r'}(t') = \{2l+2\}$. Applying Lemma 4 for time t' gives $r' = s$ and $k = l$, a contradiction with the assumption $l < k$.

2. If $r \notin V_A$, then since $c_{\{r,r'\}}(t) \in [2l, 2l+1]$, we must have $C_r(t) \subseteq [2l, 2l+2]$ (note that we must have $\min C_r(t) \geq 2l$). At time t', only the traversal count of edge $\{r, r'\}$ changes, $c_{\{r,r'\}}(t') = c_{\{r,r'\}}(t) + 1 = \min C_r(t) + 1 \in [2l+1, 2l+2]$, and we still have $C_r(t') \subseteq [2l, 2l+2]$. By the definition of set V'_A we have $r \notin V'_A$, so $C_r(t')$ fulfills the required bound with parameter $b = l$.

For vertex r' we likewise consider two possibilities: either $r' \in V_A$, or $r' \notin V_A$; in both cases, we obtain that the required bounds on $C_{r'}(t')$ are satisfied.

Finally, we consider vertex v_l (under the assumption that $l < k$, otherwise this case should be left out). Since the bounds for vertices r and r' have already been proven, we can restrict ourselves to the case of $v_l \neq r$ and $v_l \neq r'$. This means that the set of traversal counts adjacent to v_l does not change during the time interval (t, t'), i.e. $C_{v_l}(t) = C_{v_l}(t')$. Clearly, the only situation which needs some comment is when $v_l \notin V'_A$; we will show that such a case is not possible. Indeed, this would mean that $l' = l+1$ or $l' \geq k'$. If $l' = l+1$, then we would have $c_{\{r,r'\}}(t') = 2l+2$, so $c_{\{r,r'\}}(t) = 2l+1$, and since $r' \neq v_l$, we see from the inductive assumption that $r' \notin V_A$ and $C_{r'}(t) \subseteq [2l, 2l+2]$. Hence, noting that $l' = l+1$, we have $C_{r'}(t') = \{2l+2\}$, and by applying Lemma 4 for time t' we obtain $r' = s$ and $k = l$, a contradiction with the assumption $l < k$. Finally, we need to consider the case $l' \geq k'$. Then, since $k' \geq k$ and $l' \leq l+1$, we obtain $l' = k' = k = l+1$, which turns out to be a subcase of the previously considered case $l' = l+1$. □

Theorem 4. *For any graph, the cover time achieved by any* LUF *exploration is at most* $2m(D+1)$.

Proof. Consider any time moment t such that $l \geq k$. Then by Lemma 5 set V_A is empty, and for any vertex $v \in V$ we have $\max C_v(t) - \min C_v(t) \leq 2$. Let edge $\{v_a, v_b\}$ be such that $c_{\{v_a,v_b\}} \geq 2k+1$, and consider any other edge $\{u_a, u_b\}$ of the graph. Let us arbitrarily choose a shortest path $(w_1, w_2, \ldots, w_d)$, with $w_1 = u_a$ and $w_d = v_a$; obviously, $d \leq D+1$. The following relations hold: $c_{\{u_a,u_b\}}(t) \geq \min C_{w_1}(t) \geq \max C_{w_1}(t) - 2 \geq c_{\{w_1,w_2\}}(t) - 2 \geq \min C_{w_2}(t) - 2 \geq \max C_{w_2}(t) - 4 \geq \ldots \geq \max C_{w_d}(t) - 2d \geq c_{\{v_a,v_b\}} - 2d \geq 2k+1-2(D+1) = 2(k-D)-1$. So, at any time moment t such that $l \geq k > D$, each edge of the graph has been explored at least once. Notice that this is always true for the unique time moment t such that $\max C(t) = 2D+2$ and $\max C(t') = 2D+3$, and we will use this time moment t as an upper bound on cover time. Since at time $\tau = (2D+2)m+1$ we must have $\max C(\tau) > 2D+2$ by the pigeon-hole principle, we immediately obtain that $t < \tau$, and the claim follows. □

Taking into account that by Lemma 5, for any time moment t and for any vertex $v \in V$, we have $\max C_v(t) - \min C_v(t) \leq 3$, and using similar arguments as in the above proof, we obtain that at any moment of time t the following inequalities hold: $\max C(t) - \min C(t) \leq 3(D+1)$. We easily conclude that in the limit, all edges are explored with the same frequency.

Theorem 5. *For any graph, any exploration following the* LUF *strategy achieves uniform frequency on all edges,* $f_e(G) = 1/m$.

3.3 Cover Time of LUF with Modified Initial Conditions

It turns out that LUF explorations are resistant to minor perturbations, for example when the initial values of traversal count are not necessarily 0 for all edges e, but arbitrarily drawn from some range of values. We have the following theorems; details of the proofs are omitted.

Theorem 6. *For any graph, the cover time achieved by any exploration following the* LUF *strategy is* $O(m(n + p))$*, where* p *is the maximum value of edge traversal counters at time* 0.

Corollary 1. *For any graph, any exploration following the* LUF *strategy achieves uniform frequency on all edges,* $f_e(G) = 1/m$*, even when the initial values of edge traversal counts in the graph are non-zero.*

4 Final remarks

We have shown that locally fair strategies in undirected graphs can closely imitate random walks, allowing us to obtain an exploration which is fair with respect to all edges, and efficient in terms of cover time. However, the fairness criterion has to be chosen much more carefully than for symmetric directed graphs: Least-Used-First works, but Oldest-First does not.

In future work it would be interesting to study modified notions of equity, which are inspired by random walks which select the next edge to be traversed with non-uniform probability. For example, it is possible to decrease the general-case bound on the cover time of a random walk to $O(|V|^2 \log |V|)$, by applying a probability distribution which reflects the degrees of the nearest neighbors of the current node [11]. It is an open question whether a similar bound can be obtained in the deterministic sense using a derandomized strategy.

References

1. D. Aldous, J. Fill. *Reversible Markov Chains and Random Walks on Graphs.* `http://stat-www.berkeley.edu/users/aldous/RWG/book.html`, 2001.
2. R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovász, and C. Rackoff. Random walks, universal sequences and the complexity of maze problems. In: *Proceedings of the 20th Annual IEEE Symposium on the Foundations of Computer Science (FOCS'79)*, pp. 218–223, 1979.

3. M. A. Bender, A. Fernández, D. Ron, A. Sahai, and S. P. Vadhan. The power of a pebble: Exploring and mapping directed graphs. *Information and Computation*, 176(1):1–21, 2002.
4. S. N. Bhatt, S. Even, D. S. Greenberg, and R. Tayar. Traversing directed eulerian mazes. *Journal of Graph Algorithms and Applications*, 6(2):157–173, 2002.
5. J. Cooper, B. Doerr, T. Friedrich, and J. Spencer. Deterministic Random Walks on Regular Trees. In: *Proceedings of 19th ACM-SIAM Symposium on Discrete Algorithms (SODA'08)*, pp. 766–772, 2008.
6. X. Deng and C. H. Papadimitriou. Exploring an Unknown Graph. *Journal of Graph Theory*, 32(3):265–297, 1999.
7. B. Doerr and T. Friedrich. Deterministic Random Walks on the Two-Dimensional Grid. *Combinatorics, Probability and Computing*, to appear (doi:10.1017/S0963548308009589), 2009.
8. P. Fraigniaud, D. Ilcinkas, G. Peer, A. Pelc, and D. Peleg. Graph exploration by a finite automaton. *Theoretical Computer Science*, 345(2-3):331–344, 2005.
9. L. Gąsieniec, A. Pelc, T. Radzik, and X. Zhang. Tree exploration with logarithmic memory. In: *Proceedings 19th ACM-SIAM Symposium on Discrete Algorithms (SODA'07)*, pp. 585–594, 2007.
10. A. Hemmerling. *Labyrinth Problems: Labyrinth-Searching Abilities of Automata.* Teubner-Texte zur Mathematik, Vol. 114, 1989.
11. S. Ikeda, I. Kubo, and M. Yamashita. The hitting and cover times of random walks on finite graphs using local degree information. *Theoretical Computer Science*, 410(1):94–100, 2009.
12. J. Kahn, J. H. Kim, L. Lovász, and V. H. Vu. The cover time, the blanket time, and the Matthews bound. In: *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS'00)*, IEEE, pp. 467–475, 2000.
13. S. Koenig. Complexity of Edge Counting. In: Goal-Directed Acting with Incomplete Information. Technical Report CMU-CS-97-199, Carnegie Mellon University, 1997.
14. S. Koenig and R. G. Simmons. Easy and Hard Testbeds for Real-Time Search Algorithms. In: *Proceedings of the National Conference on Artificial Intelligence*, pp. 279–285, 1996.
15. M. Koucký. Universal traversal sequences with backtracking. *Journal of Computer and System Sciences*, 65(4):717–726, 2002.
16. L. Lovász. Random walks on graphs: A survey. *Bolyai Society Mathematical Studies*, 2:353–397, Budapest, 1996.
17. O. Reingold. Undirected ST-connectivity in log-space. In: *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC'05)*, pp. 376–385, 2005.
18. M. E. Saks. Randomization and derandomization in space-bounded computation. In *Proceedings of the 11th Annual IEEE Conference on Computational Complexity*, pages 128–149, 1996.
19. I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein. Smell as a Computational Resource — a Lesson We Can Learn from the Ants. In: *Proceedings of Fourth Israeli Symposium on Theory of Computing and Systems (ISTCS'96)*, pp. 219–230, 1996.
20. I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein. Distributed Covering by Ant-Robots Using Evaporating Traces. *IEEE Transactions on Robotics and Automation*, 15(5):918–933, 1999.
21. P. Winkler and D. Zuckerman. Multiple cover time. *Random Structures and Algorithms*, 9(4):403–411, 1996.
22. V. Yanovski, I. A. Wagner, and A. M. Bruckstein. A Distributed Ant Algorithm for Efficiently Patrolling a Network. *Algorithmica*, 37:165–186, 2003.